DSS 13 Microprocessor Antenna Controller

R. M. Gosline
Radio Frequency and Microwave Subsystems Section

A microprocessor based antenna controller system developed as part of the unattended station project for DSS 13 is described. Both the hardware and software top level designs are presented with a disucssion of the major problems encountered. Developments useful to related projects include a JPL standard 15 line interface using a single board computer, a general purpose parser, a fast floating point to ASCII conversion technique, and experience gained in using off board floating point processors with the 8080 CPU.

I. Introduction

The design of a microprocessor antenna controller was performed as part of the DSN unattended station development at DSS-13. The objectives were to replace the existing antenna controller (a Modcomp II) with a low cost standard 8 bit microprocessor controller similar to the microprocessor controllers used in the other automated subsystems, and to replace the current assembly language software with a high level programming language utilizing a top-down structured design. Another objective was to replace the antenna data collection system with a similar design.

II. System Description

A. Hardware

The system consists of two controllers as shown in Fig. 1. The controllers utilize only commercially available modules to simplify requirements for documentation and spares. Collecting analog and digital data from the various remote transducers and sending this data to the main controller located in the control room is the function of a data collection controller located in the antenna pedestal. The main controller collects data from local monitor points in the control room, provides operator interface, communication, display, and necessary logic and calculations for controlling the antenna. Communication to the station controller is through a Star Switch utilizing a JPL standard 15 line interface.

The remote data collection controller configuration is shown in Fig. 2. It consists of a single board computer (8080 CPU) with 8 K-bytes of onboard PROM and 4 K-bytes of RAM. The CPU drives a short haul modem through a serial port and is connected to 24 digital channels via onboard parallel ports. An external digital interface uses opto-isolator modules to convert 24 volt antenna levels to T.T.L. levels. A 32 channel differential input multiplexer and analog to digital converter provide interface to the analog channels. All input lines are protected from lightning discharges by filters as well as varistor type surge protectors. The following development modules, RAM module, floppy disk controller, communication expansion module, and dual floppy disk drives allowed software development and testing in the remote data collection controller, thus eliminating the need to use costly cross compilers and/or development systems. After checkout the program was burned into PROM and these development modules were removed, leaving only the CPU and analog to digital converter cards.

The main controller configuration is shown in Fig. 3. Two CPUs are utilized on the same Multibus. CPU 1 provides interface to the angle encoders through its parallel ports and performs most of the tracking related calculations. CPU 2 provides interface to the Star Switch through its parallel ports and the serial data from the data collection controller and modem through its serial port. CPU 2 performs the conversion of data to floating point and ASCII. It also handles the display and operator interface. In addition, CPU 2 performs the major part of the tasking logic and initiates appropriate action in the event of a fault detection.

Analog and digital interfaces are provided in a similar manner as the remote data collection controller, except that an input-output expansion module is necessary since the parallel ports of both CPUs are utilized. Numerical computations are facilitated by three floating point processors dedicated to tracking computations, digital servo-loop computations, and data conversion tasks such as binary to floating point or ASCII. A video-keyboard module provides interface to the local maintenance operator keyboard and high resolution monitor. Near real time status and performance data are displayed in the upper portion of the maintenance screen while response messages and operator inputs are displayed in the lower portion of the screen. Upper screen data that is not within expected limits, or states, is displayed for the maintenance operator in reverse video and flagged within the program for possible automatic monitor action. A communications expansion module provides interface to a printer for maintenance data logging, or for making listings of the program during development. It also provides an interface for an additional terminal which allows separate control of each CPU for debugging purposes. A digital to analog converter module provides the output drive voltages to operate the antenna servo-hydraulic system. As with the remote data collection controller, software was developed in the controller using a floppy disk controller and two disk drives.

B. Software

The program is written in the PL/M (an Intel trademark) language. Currently the main controller program contains about 10,000 lines of source code with 200 lines of assembly language for high speed peripheral drivers. The compiled program requires 54 K-bytes including data storage. For comparison, the minicomputer program contained 2,300 lines of FORTRAN and 4,600 lines of assembly language requiring 30 K-words of memory (a Modcomp word is 2 bytes).

The top level flow chart for the remote data collection controller is shown in Fig. 4. All of the flowcharts shown have been simplified for illustration purposes, and some modules may actually represent a combination of several. After an initialization module, the program enters an endless loop containing the routines REMDAT, which inputs data from the transducers, and REMSEND, which formats, adds a checksum, and sends the data over the serial link to the main controller. A timer on board the CPU provides a 50 ms interrupt which is divided modulo 4 so that the main loop is executed every 200 ms. The serial line operates at 9,600 baud. The data is formatted in two 73 character blocks of ASCII characters which can be displayed directly on a local maintenance terminal if desired for interface verification.

The main controller top level software is shown in Fig. 5. Both CPUs begin execution of the same code from exactly the same memory location. Both have the ability to interrogate themselves to find out which CPU they are and jump to their separate initialization routines. From there each enters a separate endless background loop except for the servicing of their interrupts.

The level 2 flowchart for CPU 1 is shown in Fig. 6. The background modules perform the following functions:

CHKSUM	Verifies that the code has not changed
SCHEDULER	Looks at current time and initiates any tasks that may be scheduled
CHKSTAK	Generates a warning message if the stack is getting too large

There are three interrupts used with CPU 1. Interrupts 3 and 4 are driven from the station clock at rates of 50 Hz and 1 Hz, respectively, which control all the time dependent motion of the antenna. Interrupt 2 is used to read a character of serial time from the station clock. The station clock is read initially to establish time. Thereafter the antenna clock is incremented in the 1 Hz interrupt routine. The 50 Hz routine counts each interrupt to insure exact synchronization with the 1 Hz interrupt. If the count is not correct or if the antenna time does not match station time, an error message is generated. The count is also used modulo 5 to call the following modules every 100 ms:

READOUTS	Inputs current position from the angle encoders
UPDATE	Updates the angle commands to the current time
SERVO	Closes the digital servo loop (type 1 or type 2 depending on start up conditions)
D2AOUT	Updates the current output drive voltage

After the interrupt 4 routine (1 Hz) steps the clock, the following modules are called:

ANAACTCNT	Decrements the counters which delay action on analog channels until after specified delays
TIMERS	Decrements action timers
CMPRATES	Computes the angle rates by differencing the position readouts

There are three tracking modes and an idle mode. An azimuth elevation mode allows pointing to a given azimuth elevation angle pair. A sidereal rate mode is used for tracking objects at sidereal rate. A three day fit mode is used for tracking objects such as spacecraft that are near sidereal rate and uses a second order fit of three values of right ascension and declination. Depending on the mode, one of the routines AEPRED, SID, or FIT3 is called to compute the current pointing angle in azimuth elevation coordinates. Three more routines are called:

CHKPRELIM	Checks the physical movement limits of the antenna and stops motion if they are about to be exceeded
REFC	Computes a refraction correction using the corrected Berman model
WINDAVG	Calculates the average wind based on the last 2 minutes; if the average wind exceeds its limit the antenna will automatically stow

The CPU 2 level 2 software is shown in Fig. 7. The background routines perform the following functions:

SPEEDSELECT	Selects either high or low speed mode depending on the distance to the desired point. An azimuth distance greater than 10 degrees or elevation greater than 3 degrees will select high speed.
CONVERT	Inputs the control room monitor channels and converts all channels (including those from the data collection controller) to floating point and ASCII. This module also checks limits and writes data to the DATAOUT buffers. If monitor action is required, flags are set for the TASKREQ module.
MONACT	Performs the actual tasks selected by the logic in TASKREQ as the result of a monitor action.
TSKACT	Performs the actual tasks selected by the logic in TASKREQ as the result of an operator or scheduler input.
DATAOUT	Outputs data from the screen, Star Switch or printer buffers.

Four interrupt routines handle the receipt of a character from the data collection controller, a Star Switch message, an operator keyboard character, or a Star Switch message time out.

Five help menus are available at the local display, the station controller display, or the remote display in the NOCC to provide the operator with assistance in remembering the commands available and their syntax. They are shown in Fig. 8. All operator and monitor events are logged on the station printer as well as reported through the Star Switch for eventual recording on magnetic media. A typical printout of a Pioneer 8 track is shown in Fig. 9.

III. Related Applications

This project produced several results that have application to other projects. A single board computer (available from Intel or National) was configured to function as a JPL standard 15 line interface port. The module and software are applicable to any Multibus microprocessor and require no external circuitry other than a cable adapter from a ribbon to an MS type connector. This module is currently being installed in transmitter and maser controllers replacing the JPL designed quad 15 line interface requiring two Multibus slots.

A general purpose parser was written to handle multiple line inputs of fields containing alphanumeric, integer, and fixed point values. The design accepts any nonalphabetic or nonnumeric character as a delimiter and considers consecutive delimiters as a single delimiter, thus providing free form inputs. An ampersand character (&) is used to indicate the end of a physical line and the continuation of a logical line for long line inputs.

Because of the requirement to update the local display with angles and analog data in near real time, a technique for fast conversion of floating point numbers to ASCII was developed. The technique is to separate the integer and fraction parts with the floating point processor into two 10 bit integers (the fractional part is scaled by 1,000). An assembly language routine then does two table look ups, using five bits at a time, to get 4 bit binary coded decimal values. Use is made of the 8080 BCD addition capability to combine the result. Conversion to

ASCII is then simply done by addition of a hexadecimal 30 to each BCD character. A floating point angle can be converted in about 300 microseconds. A faster technique, of course, is to simply use a table look up on all 10 bits. This was the original design before memory constraints required a different technique.

Although not completed at this time, two features are being included to be used with the SETI project for performing special scanning requirements. One is a raster scan pattern specified by overall size and scan rates. The second is a general three point fit capability to allow fitting a second order curve to any three points of right ascension-declination or azimuth-elevation. The current three day fit requires the points at zero hours on successive days. The more general method will allow the points to occur at unequal time intervals down to one second.

A great deal of experience has been gained configuring a maximum loaded 8080 based system with multiple processors. Difficulty was encountered in debugging problems relating to critical timing relationships between the two CPUs. Present generation logic analyzers lack the capability of triggering on combined events in each CPU and back-tracing either CPU or the bus.

Another inconvenience is the inability of the 8080 (and most other CPUs including 16 bit devices) to perform bit operations. The least addressable element is a byte. This means that to change a single bit in memory, an input-output port, or interrupt mask requires reading the byte, changing the desired bit, and storing the byte. If more than one CPU or an interrupt routine attempts to alter a bit in the same byte one of the changes may not be made. The Z80 CPU does allow bit operations, which is desirable for control applications.

All software was developed in the controllers or similar microprocessor computers. While this proved convenient from the standpoint of always having an editor or compiler available, the time required for making changes in several modules, compiling, relinking, listing, and reloading typically was 4 hours or more. Much of the debugging and patching was performed at the machine code level however, as one of the advantages of PL/M is that the source statements are easily traceable to the machine level code.

Several undesirable features of the floating point processors were encountered and should be given consideration when choosing components for future systems. Stack oriented processors cannot be shared by multiple users (such as interrupt routines or other CPUs) if their state cannot be restored to the state at the time of the interrupt. The capability to save the processor stack and restore it existed, but the status register could not be restored. It would also be useful in many applications to have access to the remainder following a fixed point division for modulo operations. Another annoying feature of the floating point processors was, if given operands that were out of range, an unpredictable result was generated rather than a closest approximation. In the refraction algorithm for example, it is necessary to evaluate small powers of e. If the argument is near zero, a very large result is obtained instead of unity. In time critical control applications, where it is not possible to check the range of all operands before computation, a closest approximation would be desirable.

The PL/M compiler available for the 8080 does not support floating point directly. Operands must be pushed onto the floating point processor stack, an operation code supplied, the result taken off the stack and stored. The execution time for getting numbers in and out of the stack is longer than many of the floating point operations. Some calculations can be chained to minimize this overhead but clever manipulation of a stack is one of the tasks that one hopes to avoid by using a high level language in the first place. The PL/M86 compiler for the 8086 CPU using an onboard floating point processor avoids this and allows floating point operations to be written algebraic form.

IV. Current Status

The system has been operational for several months for Pioneer 8 tracks and is now being used for all DSS 13 tracking operations. All features are not complete however, and work is continuing in the following areas: The general three point fit algorithm and raster scan for SETI are currently being implemented. The scheduler module to allow scheduled tasks to occur at a specified time and the capability of reading commonly used source predicts from an on line file are not complete at this time.

V. Conclusion

While the objectives of this effort have been achieved, it is clear that this represents a data point on the maximum capability of an eight bit microprocessor based system. Limits of memory size, execution speed, module space, and general complexity were encountered. All of these limits could be corrected by use of one of the more powerful 16 bit processors that have become available since the start of this project. The software is transferable to a 16 bit processor with minor modifications.

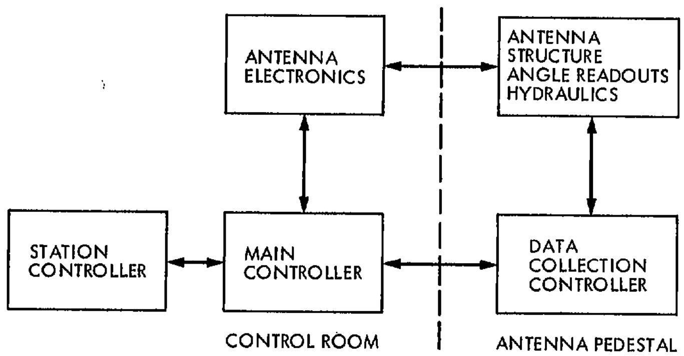

Fig. 1. DSS 13 antenna controller functional block diagram

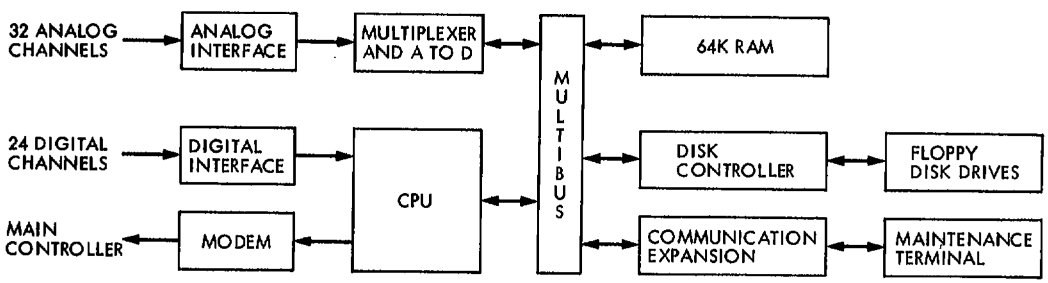

Fig. 2. Data collection controller configuration

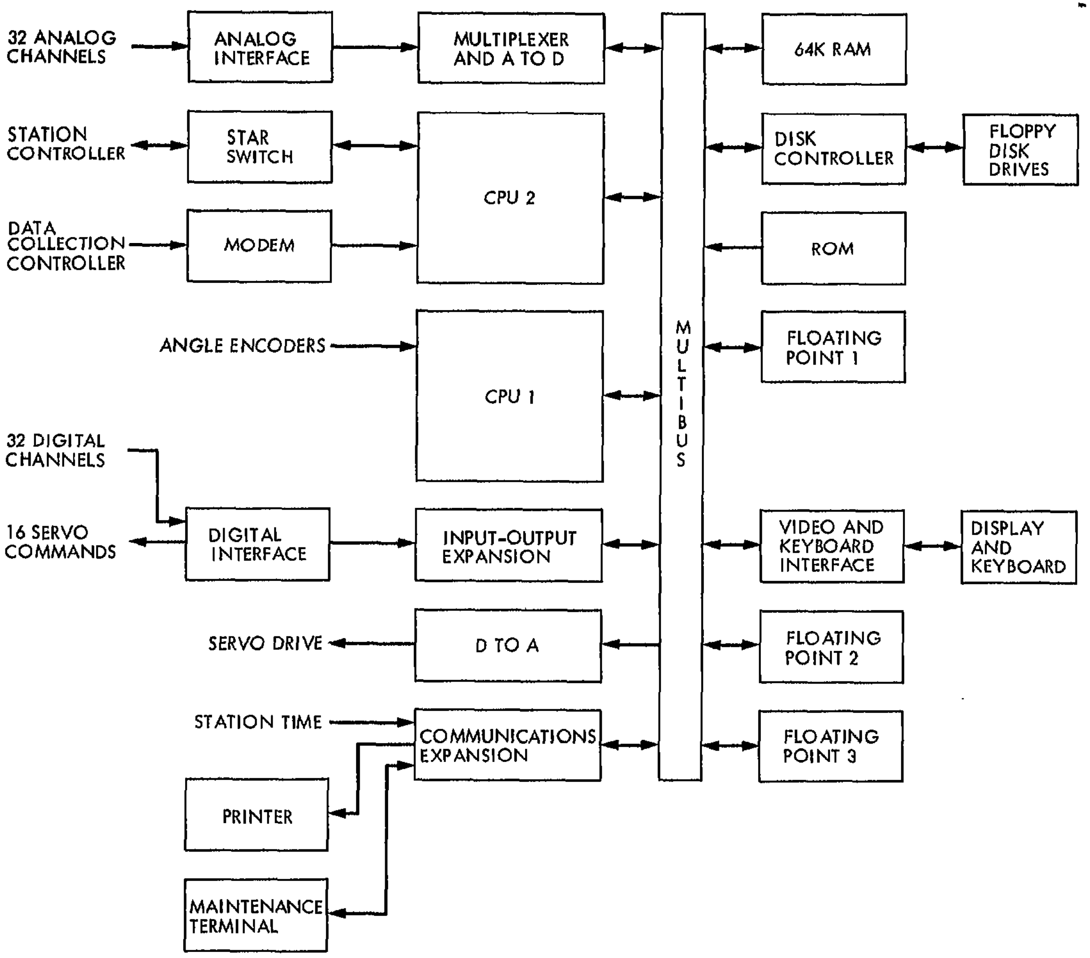

Fig. 3. Main controller configuration

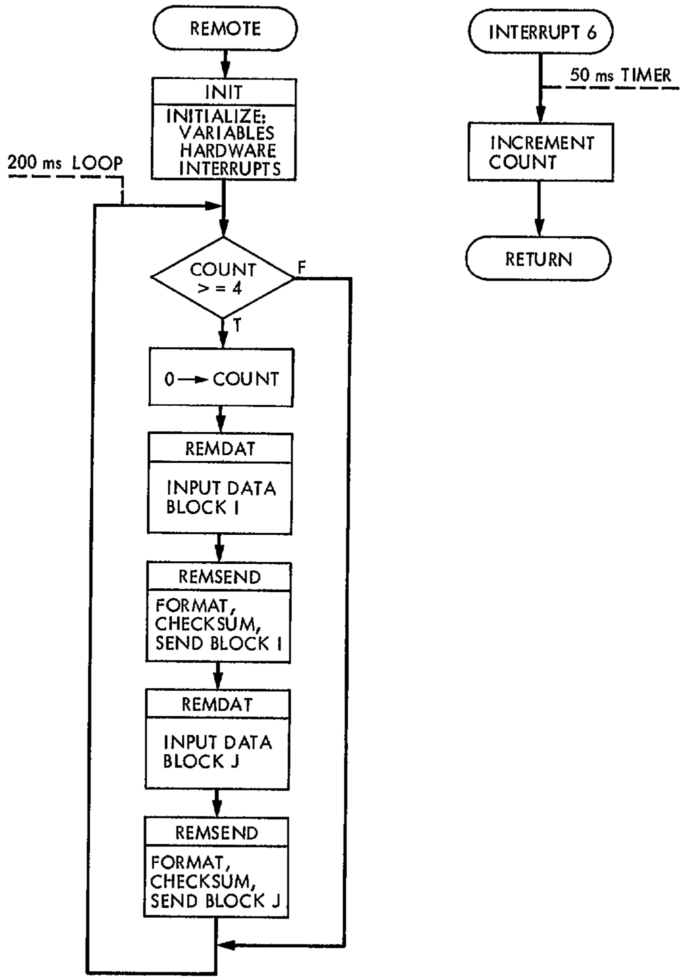

Fig. 4. Data collection controller top level software

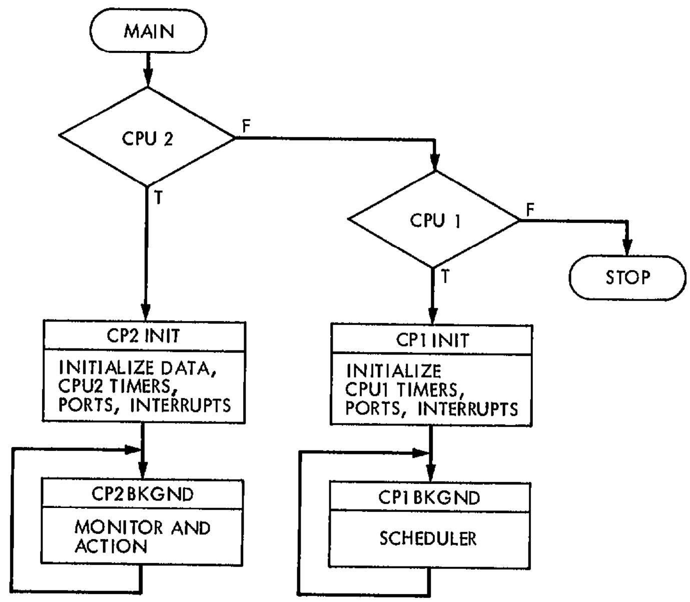

Fig. 5. Main controller top level software

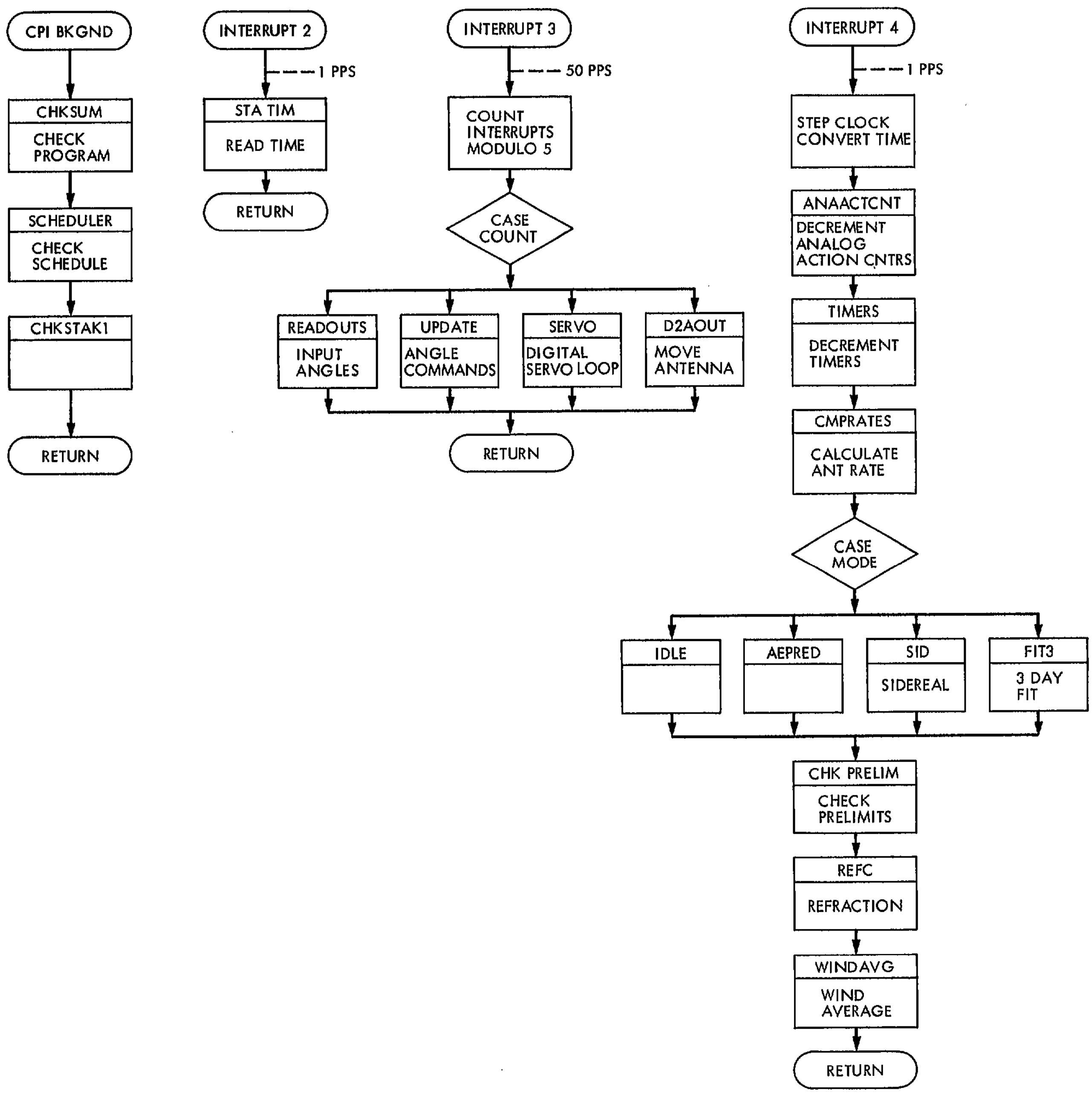

Fig. 6. CPU 1 level 2 software

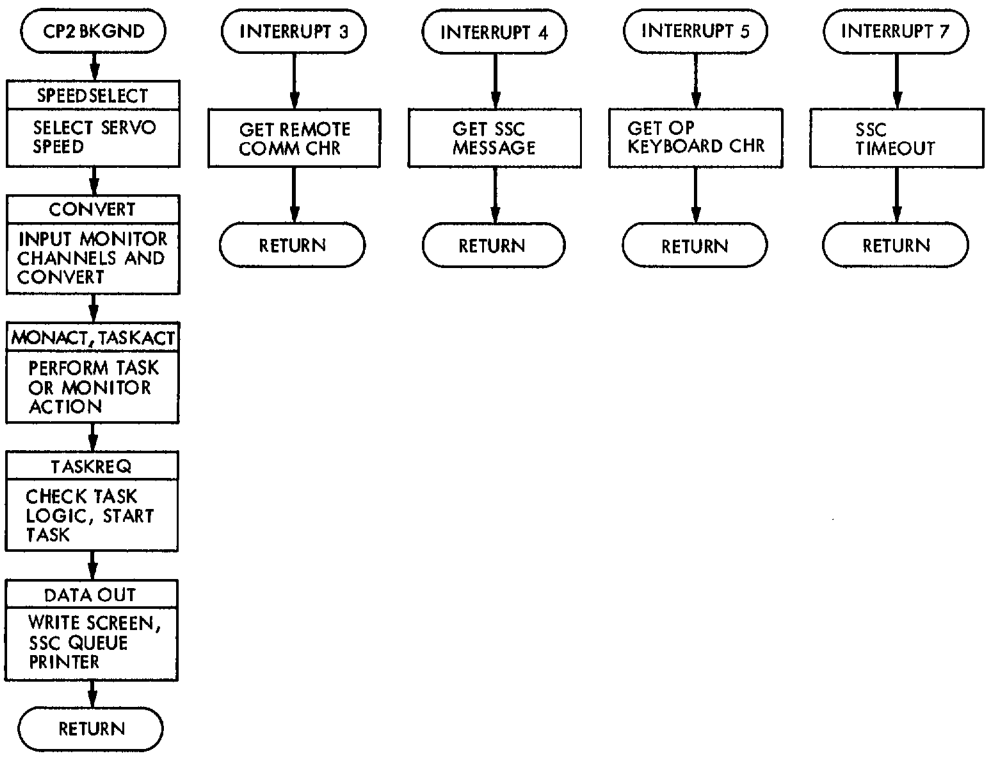

Fig. 7. CPU 2 level 2 software

```
HELP
HELP MENUS - TYPE:
 HELP 0 - station level commands
 HELP 1 - cnf formats
 HELP 2 - str formats
 HELP 3 - maint mode commands

HELP 0
ANTENNA CONTROLLER STATION LEVEL COMMANDS:
 INIT    : TURNS ON ELECTRONICS AND HYDRAULICS.
 CNF     : INDICATES START OF SCHEDULE BLOCK DATA.
 OFF     : STOPS ANTENNA, TURNS ELECTRONICS AND HYDRAULICS OFF.
 OPERATE: EXAMINE AND INITIATE ACTION SPECIFIED IN SCHEDULE BLOCK
 STOW    : ANTENNA TO STOW POSITION; ELECTRONICS, HYDRAULICS OFF.
 STOP    : SLOWS ANTENNA, APPLIES BRAKES.
 STR     : REPORT STATUS OR DATA.
 CLEAR   : RESET ANTENNA LOCKOUT AFTER A FAULT
 HELP    : DISPLAY HELP MENU

HELP 1
ANTENNA CONTROLLER CNF MENU
 SID  <name><angles>
 3DAY <name><3 sets of angles>
 AZEL <angles><wrap-up cmd>
 DFOF <angles>, [direct offsets]
 AEOF <angles>, [az/cos el]
 CLDO, [clear DFOF]
 CLPO, [clears AEOF, RDOF]
 RDOF <angles>, [ra, dec offsets]
 RAEO <angle rates>, [rate offsets]
 CLRO, [clear rate offsets]

HELP 2
ANTENNA CONTROLLER STR MENU
 STR  CNF  : CURRENT CNF BLOCK
 STR  ANA  : ANALOG DATA
 STR  DIG  : DIGITAL DATA
 STR  ANG  : ANGLE DATA
 STR  ALL  : ALL DATA
 STR  TANG : TIME, ANGLES
 STR       : SUBSYSTEM STATUS

HELP 3
MAINTENANCE COMMANDS
   10-SET BRAKE AZ        30-SET BRAKE EL
   11-RELEASE BRAKE AZ    31-RELEASE BRAKE EL
   12-HIGH SPEED AZ       32-HIGH SPEED EL
   13-LOW SPEED AZ        33-LOW SPEED EL
   14-ELECTRONICS ON      34-ELECTRONICS OFF
   15-PUMPS ON            35-PUMPS OFF
    7-INCR RATE AZ         9-INCR RATE EL
    8-DECR RATE AZ         0-DECR RATE EL
   19-IGNORE LIMITS AZ    39-IGNORE LIMITS EL
   20-SET LIMITS AZ       40-SET LIMITS EL
   21-PRINT CURRENT INCREMENTAL RATES
   23-PRINT RATE INCREMENT
```

Fig. 8. Help menus

```
053 15:40:58 REM
053 15:42:28 F5;INIT;
053 15:42:30 INITIALIZING



053 15:42:42 PUMP 75HP START RLY = ON
053 15:43:13 PUMP 125HP START RLY= ON
053 15:43:28 F5;CNF;
053 15:43:53 F5;3DAY PN8 246.659 -21.586
053 15:44:04 INIT COMPLETE
053 15:44:11 F5;247.584 -21.720
053 15:44:29 F5;248.510 -21.849
053 15:44:36 F5;OPER;
053 15:44:54 MOVING TO POINT
053 15:46:19 ON SOURCE
053 15:46:20 OPER COMPLETE
053 15:49:03 F5;STR;
053 15:49:03 ANTENNA IS UNDER REMOTE CONTROL
053 15:49:03 AZ =  206.447 EL =   28.389
053 15:49:03 AVG WIND =  9.29 MPH
053 15:49:17 F5;STR;
053 15:49:17 ANTENNA IS UNDER REMOTE CONTROL
053 15:49:17 AZ =  206.507 EL =   28.366
053 15:49:17 AVG WIND =  9.29 MPH
053 15:49:56 F5;DFOF .087 -.01
053 16:09:02 F5;STR;
053 16:09:02 ANTENNA IS UNDER REMOTE CONTROL
053 16:09:02 AZ =  211.271 EL =   26.411
053 16:09:02 AVG WIND =  10.7 MPH
053 16:19:01 F5;STR;
053 16:19:01 ANTENNA IS UNDER REMOTE CONTROL
053 16:19:01 AZ =  213.548 EL =   25.322
053 16:19:01 AVG WIND =  12.0 MPH
053 16:24:22 F5;STR;
053 16:24:22 ANTENNA IS UNDER REMOTE CONTROL
053 16:24:22 AZ =  214.738 EL =   24.712
053 16:24:22 AVG WIND =  11.8 MPH
053 16:24:52 F5;STR ANA;
053 16:24:52 A32 EL LS MOTR DIF PRESS -17.6  PSI
053 16:24:52 A33 EL HS MOTR DIF PRESS  13.2  PSI
053 16:24:52 A34 EL SYSTEM PRESS       2880  PSI
053 16:24:52 A35 EL HS PRESS           82.0  PSI
053 16:24:52 A36 EL LS PRESS           2630  PSI
053 16:24:52 A37 AZ HS MOTR DIF PRESS -11.7  PSI
053 16:24:52 A38 AZ LS MOTR DIF PRESS  341.  PSI
053 16:24:52 A39 AZ HS SYSTEM PRESS    2190  PSI
053 16:24:52 A40 AZ LS SYSTEM PRESS    1810  PSI
053 16:24:52 A41 WIND SPEED SW TOWER   16.6  MPH
053 16:24:52 A42 WIND SPEED SE TOWER   14.9  MPH
053 16:24:52 A43 WIND AZ SW TOWER      349.  DEG
053 16:24:52 A44 WIND AZ SE TOWER      302.  DEG
053 16:24:52 A45 PUMP VOL 75HP LEFT    27.2  GPM
```

Fig. 9. Typical operation record